In [ ]:
```
# Read the Feature engineering DataSet

#!wget --header="Host: doc-0s-0c-docs.googleusercontent.com" --header="User-Agent
```

In [ ]:
```
# Import all The Library

import pandas as pd
import numpy as np
import lightgbm as lgb
from tqdm import tqdm
import joblib
import random
```

In [ ]:
```
# Read The Data

df_m = pd.read_pickle("Data_m5.pkl")
```

In [ ]:
```
# Size of DataFrame

df_m.shape
```

Out[4]: (27023821, 47)

In [ ]:
```
df_m.head()
```

Out[5]:

	id	item_id	dept_id	cat_id	store_id	state_id	d	sales	da
0	HOBBIES_1_001_CA_1_evaluation	0	0	0	0	0	d_1050	0.0	201 12-1
1	HOBBIES_1_002_CA_1_evaluation	1	0	0	0	0	d_1050	0.0	201 12-1
2	HOBBIES_1_003_CA_1_evaluation	2	0	0	0	0	d_1050	0.0	201 12-1
3	HOBBIES_1_004_CA_1_evaluation	3	0	0	0	0	d_1050	0.0	201 12-1
4	HOBBIES_1_005_CA_1_evaluation	4	0	0	0	0	d_1050	0.0	201 12-1

In [ ]:
```
# Fill Zero Values

for c in [c for c in df_m.columns.tolist() if 'rm_diff_price_' in c]:
    df_m[c].fillna(0, inplace=True)
```

In [ ]:
```
# Get integer value in d column    ex d_1 , d_2   ---->>>   1  ,1

df_m['d'] = df_m['d'].apply(lambda a: a.split('_')[1]).astype(np.int16)
```

In [ ]:
```
# Training DataSet

df = df_m[df_m['d']<=1941]
```

In [ ]:
```
df.head()
```

Out[9]:

	id	item_id	dept_id	cat_id	store_id	state_id	d	sales	date
0	HOBBIES_1_001_CA_1_evaluation	0	0	0	0	0	1050	0.0	2013-12-13
1	HOBBIES_1_002_CA_1_evaluation	1	0	0	0	0	1050	0.0	2013-12-13
2	HOBBIES_1_003_CA_1_evaluation	2	0	0	0	0	1050	0.0	2013-12-13
3	HOBBIES_1_004_CA_1_evaluation	3	0	0	0	0	1050	0.0	2013-12-13
4	HOBBIES_1_005_CA_1_evaluation	4	0	0	0	0	1050	0.0	2013-12-13

In [ ]:
```
df.shape
```

Out[10]:
```
(26170101, 47)
```

In [ ]:
```
# Categorical Feature

cat_feats = (['item_id','store_id', 'cat_id', 'state_id','dept_id']
             + ["event_type_1", "event_type_2"]
             + ['wday', 'month', 'snap_CA', 'snap_TX', 'snap_WI'])
```

In [ ]:
```
# Columns i want to Remove

useless_cols = ["id", "date", "sales","d", "wm_yr_wk", "weekday", "sell_price",'
```

In [ ]:
```
train_cols = df.columns[~df.columns.isin(useless_cols)]
```

In [ ]:
```
# Chosse 500 Random Days For Cross Validation DataSet

days_val = random.choices(df['d'].unique().tolist(), k=500)
```

In [ ]:
```
# Training Data
X_train = df[df['d'].isin(days_val)==False][train_cols]
# Training Target Variable
y_train = df[df['d'].isin(days_val)==False]["sales"]
```

In [ ]:
```
# Cross Validation Data
X_val = df[df['d'].isin(days_val)==True][train_cols]
# Cross Validation Target Variable
y_val = df[df['d'].isin(days_val)==True]["sales"]
```

Modeling Part

In [ ]:
```
%%time

learning_rate1 = [0.09,.05,.15]
number_of_Leave = [32,16,8]
lambda_l2 = [0.1,0.2,0.3]
min_data_in_leaf1 = [50,30,20]




for kk in tqdm(range(3)):

    print("Prediction Number  Started ",kk)

    # Instilaize All Variables
    lr1 = learning_rate1[kk]
    leave1 = number_of_Leave[kk]
    reg1 = lambda_l2[kk]
    leaf_data = min_data_in_leaf1[kk]

    params = {
        "objective" : "poisson",
        "metric" :"rmse",
        "learning_rate" : lr1,
        "sub_feature" : 0.9,
        "sub_row" : 0.75,
        "bagging_freq" : 1,
        "lambda_l2" : reg1,
        'verbosity': 1,
        'num_iterations' : 2000,
        'num_leaves': leave1,
        "min_data_in_leaf": leaf_data,
    }

    print("*"*50)
    print(params)
    print(" "*50)


    train_data = lgb.Dataset(X_train, label = y_train, categorical_feature=cat_f
    valid_data = lgb.Dataset(X_val, label = y_val, categorical_feature=cat_feats

    m_lgb = lgb.train(params, train_data, valid_sets = [train_data, valid_data],
                verbose_eval=20, early_stopping_rounds=30)


    model_name = 'lgb'+'_Model_' +str(kk) +'.pkl'
    # save model
    joblib.dump(m_lgb, model_name)



    feature_imp = pd.DataFrame({'Value':m_lgb.feature_importance(),'Feature':X_t
    feature_imp = feature_imp.sort_values(by='Value', ascending=False).reset_ind
    display(feature_imp.head(20))
```

```
    print("*"*50)
    print(" "*50)
```

```
  0%|          | 0/3 [00:00<?, ?it/s]/usr/local/lib/python3.7/dist-packages/l
ightgbm/engine.py:118: UserWarning: Found `num_iterations` in params. Will us
e it instead of argument
  warnings.warn("Found `{}` in params. Will use it instead of argument".forma
t(alias))
/usr/local/lib/python3.7/dist-packages/lightgbm/basic.py:1205: UserWarning: U
sing categorical_feature in Dataset.
  warnings.warn('Using categorical_feature in Dataset.')

Prediction Number  Started  0
**************************************************
{'objective': 'poisson', 'metric': 'rmse', 'learning_rate': 0.09, 'sub_featur
e': 0.9, 'sub_row': 0.75, 'bagging_freq': 1, 'lambda_l2': 0.1, 'verbosity':
1, 'num_iterations': 2000, 'num_leaves': 32, 'min_data_in_leaf': 50}


/usr/local/lib/python3.7/dist-packages/lightgbm/basic.py:762: UserWarning: ca
tegorical_feature in param dict is overridden.
  warnings.warn('categorical_feature in param dict is overridden.')
```

Testing Purpose

In [ ]:
```
# For Prediction Purpose i am using This DataFrame

dt = df_m[df_m['d']>=1871]
```

In [ ]:
```
dt.head()
```

Out[ ]:

	id	item_id	dept_id	cat_id	store_id	state_id	d	sales
23974822	HOBBIES_1_001_CA_1_evaluation	0	0	0	0	0	1871	0.0
23974823	HOBBIES_1_001_CA_1_evaluation	0	0	0	0	0	1872	0.0
23974824	HOBBIES_1_001_CA_1_evaluation	0	0	0	0	0	1873	1.0
23974825	HOBBIES_1_001_CA_1_evaluation	0	0	0	0	0	1874	1.0
23974826	HOBBIES_1_001_CA_1_evaluation	0	0	0	0	0	1875	1.0

In [ ]:
```
dt.shape
```

Out[ ]:
```
(3018510, 47)
```

In [ ]:
```
def create_fea(dt):

    # Create 2 Lags of 7 Days And 28 Days
    lags = [7, 28]
    lag_cols = [f"lag_{lag}" for lag in lags ]
    for lag, lag_col in zip(lags, lag_cols):
        dt[lag_col] = dt[["id","sales"]].groupby("id")["sales"].shift(lag)


    # Create 4 Rolling Mean
    wins = [7, 28]
    for win in wins :
        for lag,lag_col in zip(lags, lag_cols):
            dt[f"rmean_{lag}_{win}"] = dt[["id", lag_col]].groupby("id")[lag_col
```

In [ ]:
```
# Duration You Want to Predict
h = 28

# Days For Lag Calculation purpose
max_lags = 70

# First Day Of Prediction
fday = datetime(2016,5, 23)
fday
```

Out[ ]:
```
datetime.datetime(2016, 5, 23, 0, 0)
```

In [ ]:
```
# Predict Sales for 28 Days

for tdelta in range(0, 28):
    day = fday + timedelta(days=tdelta)
    print(day)
    tst = dt[(dt.date >= day - timedelta(days=max_lags)) & (dt.date <= day)]
    create_fea(tst)
    tst = tst.loc[tst.date == day , train_cols]
    te.loc[te.date == day, "sales"] = m_lgb.predict(tst)
    print("Prediction Compltete ",tdelta)
    del(tst)
```

```
2016-05-23 00:00:00

/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:5: SettingWithCopy
Warning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/sta
ble/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pyd
ata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-c
opy)
  """
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:10: SettingWithCop
yWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/sta
ble/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pyd
ata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-c
opy)
  # Remove the CWD from sys.path while we load stuff.
/usr/local/lib/python3.6/dist-packages/pandas/core/indexing.py:1743: SettingWit
hCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/sta
ble/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pyd
ata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-c
opy)
  isetter(ilocs[0], value)


Prediction Compltete  0
2016-05-24 00:00:00
Prediction Compltete  1
2016-05-25 00:00:00
Prediction Compltete  2
2016-05-26 00:00:00
Prediction Compltete  3
2016-05-27 00:00:00
Prediction Compltete  4
2016-05-28 00:00:00
Prediction Compltete  5
```

```
2016-05-29 00:00:00
Prediction Compltete  6
2016-05-30 00:00:00
Prediction Compltete  7
2016-05-31 00:00:00
Prediction Compltete  8
2016-06-01 00:00:00
Prediction Compltete  9
2016-06-02 00:00:00
Prediction Compltete  10
2016-06-03 00:00:00
Prediction Compltete  11
2016-06-04 00:00:00
Prediction Compltete  12
2016-06-05 00:00:00
Prediction Compltete  13
2016-06-06 00:00:00
Prediction Compltete  14
2016-06-07 00:00:00
Prediction Compltete  15
2016-06-08 00:00:00
Prediction Compltete  16
2016-06-09 00:00:00
Prediction Compltete  17
2016-06-10 00:00:00
Prediction Compltete  18
2016-06-11 00:00:00
Prediction Compltete  19
2016-06-12 00:00:00
Prediction Compltete  20
2016-06-13 00:00:00
Prediction Compltete  21
2016-06-14 00:00:00
Prediction Compltete  22
2016-06-15 00:00:00
Prediction Compltete  23
2016-06-16 00:00:00
Prediction Compltete  24
2016-06-17 00:00:00
Prediction Compltete  25
2016-06-18 00:00:00
Prediction Compltete  26
2016-06-19 00:00:00
Prediction Compltete  27
```

In [ ]:
```
te.head()
```

Out[62]:

	id	item_id	dept_id	cat_id	store_id	state_id	d	sales
23974822	HOBBIES_1_001_CA_1_evaluation	0	0	0	0	0	1871	0.0
23974823	HOBBIES_1_001_CA_1_evaluation	0	0	0	0	0	1872	0.0
23974824	HOBBIES_1_001_CA_1_evaluation	0	0	0	0	0	1873	1.0
23974825	HOBBIES_1_001_CA_1_evaluation	0	0	0	0	0	1874	1.0
23974826	HOBBIES_1_001_CA_1_evaluation	0	0	0	0	0	1875	1.0

In [ ]:
```
# Filter Last 28 Days

x = pd.pivot_table(te, index='id', values='sales', columns='d').iloc[:,-28:]
x
```

Out[ ]:

d	1942	1943	1944	1945	1946	1947	
id							
FOODS_1_001_CA_1_evaluation	0.810059	0.717285	0.701172	0.752930	0.825684	0.916992	0
FOODS_1_001_CA_2_evaluation	0.750977	0.710938	0.634766	0.622559	0.768066	1.092773	0
FOODS_1_001_CA_3_evaluation	0.890625	0.825684	0.785645	0.752441	1.304688	1.141602	1
FOODS_1_001_CA_4_evaluation	0.269287	0.267822	0.291016	0.300781	0.314941	0.327148	0
FOODS_1_001_TX_1_evaluation	0.501953	0.616211	0.550293	0.630859	0.778809	0.753418	0
...	...	...	...	...	...	...	
HOUSEHOLD_2_516_TX_2_evaluation	0.202637	0.184570	0.177612	0.172241	0.193848	0.202148	0
HOUSEHOLD_2_516_TX_3_evaluation	0.173462	0.158936	0.153320	0.154907	0.182495	0.207642	0
HOUSEHOLD_2_516_WI_1_evaluation	0.117126	0.108398	0.109375	0.119751	0.181396	0.186768	0
HOUSEHOLD_2_516_WI_2_evaluation	0.122437	0.112244	0.112183	0.114197	0.132202	0.127441	0
HOUSEHOLD_2_516_WI_3_evaluation	0.105896	0.101257	0.101196	0.103271	0.114258	0.097900	0

30490 rows × 28 columns

In [ ]:
```
x.to_csv("sub.csv",index='False')
```

In [ ]:
```
a1 = pd.read_csv("sub.csv")
a1
```

Out[66]:

	id	1942	1943	1944	1945	1946	1947	1
0	FOODS_1_001_CA_1_evaluation	0.84860	0.73730	0.73830	0.7550	0.8076	0.93200	0.88
1	FOODS_1_001_CA_2_evaluation	0.76660	0.71400	0.68500	0.6343	0.7812	1.09400	0.89
2	FOODS_1_001_CA_3_evaluation	0.81050	0.71300	0.71630	0.6836	1.2290	1.04600	1.08
3	FOODS_1_001_CA_4_evaluation	0.26560	0.26700	0.28320	0.3018	0.2957	0.31470	0.31
4	FOODS_1_001_TX_1_evaluation	0.47880	0.67870	0.54700	0.5660	0.6772	0.65770	0.80
...	...	...	...	...	...	...	...	
30485	HOUSEHOLD_2_516_TX_2_evaluation	0.20750	0.19180	0.17910	0.1743	0.2048	0.22470	0.24
30486	HOUSEHOLD_2_516_TX_3_evaluation	0.17880	0.15820	0.14900	0.1486	0.1929	0.22390	0.31
30487	HOUSEHOLD_2_516_WI_1_evaluation	0.12220	0.11430	0.11536	0.1216	0.1871	0.20980	0.20
30488	HOUSEHOLD_2_516_WI_2_evaluation	0.12260	0.11690	0.11890	0.1119	0.1267	0.13750	0.12
30489	HOUSEHOLD_2_516_WI_3_evaluation	0.11237	0.10693	0.10284	0.1047	0.1129	0.10803	0.11

30490 rows × 29 columns

In [79]:
```
sub = pd.read_csv('sample_submission.csv', usecols=['id'])
sub
```

Out[79]:

	id
0	HOBBIES_1_001_CA_1_validation
1	HOBBIES_1_002_CA_1_validation
2	HOBBIES_1_003_CA_1_validation
3	HOBBIES_1_004_CA_1_validation
4	HOBBIES_1_005_CA_1_validation
...	...
60975	FOODS_3_823_WI_3_evaluation
60976	FOODS_3_824_WI_3_evaluation
60977	FOODS_3_825_WI_3_evaluation
60978	FOODS_3_826_WI_3_evaluation
60979	FOODS_3_827_WI_3_evaluation

60980 rows × 1 columns

In [ ]:
```
sub = sub.merge(a1, on='id', how='left')
sub
```

Out[68]:

	id	1942	1943	1944	1945	1946	1947	1948	19
0	HOBBIES_1_001_CA_1_validation	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
1	HOBBIES_1_002_CA_1_validation	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
2	HOBBIES_1_003_CA_1_validation	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
3	HOBBIES_1_004_CA_1_validation	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
4	HOBBIES_1_005_CA_1_validation	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
...	...	...	...	...	...	...	...	...	
60975	FOODS_3_823_WI_3_evaluation	0.5010	0.5205	0.5350	0.5020	0.5230	0.6855	0.6787	0.55
60976	FOODS_3_824_WI_3_evaluation	0.2323	0.2384	0.2540	0.3364	0.3499	0.3796	0.3757	0.32
60977	FOODS_3_825_WI_3_evaluation	0.8228	0.6220	0.6377	0.6045	0.6826	0.7610	0.8887	0.71
60978	FOODS_3_826_WI_3_evaluation	1.1470	1.2930	1.1380	1.1330	1.2000	1.3890	1.2960	1.18
60979	FOODS_3_827_WI_3_evaluation	1.3790	1.1210	0.9310	1.2430	1.4160	1.5300	1.4630	1.04

60980 rows × 29 columns

In [ ]:
```
sub = sub.dropna()
sub
```

Out[ ]:

	id	1942	1943	1944	1945	1946	1947	1948	19
30490	HOBBIES_1_001_CA_1_evaluation	0.9010	0.8433	0.8200	0.9210	1.1490	1.2790	1.2370	1.07
30491	HOBBIES_1_002_CA_1_evaluation	0.2512	0.2673	0.2458	0.2461	0.2908	0.3926	0.3003	0.22
30492	HOBBIES_1_003_CA_1_evaluation	0.4856	0.4438	0.4880	0.5390	0.6690	0.7750	0.6590	0.57
30493	HOBBIES_1_004_CA_1_evaluation	1.8310	1.5380	1.5770	1.6790	1.8230	2.0430	2.3090	2.01
30494	HOBBIES_1_005_CA_1_evaluation	1.4600	1.3020	1.1340	1.1810	1.4880	1.4910	1.2940	1.25
...	...	...	...	...	...	...	...	...	
60975	FOODS_3_823_WI_3_evaluation	0.4850	0.5100	0.5146	0.4760	0.5176	0.6133	0.5537	0.54
60976	FOODS_3_824_WI_3_evaluation	0.2524	0.2502	0.2664	0.3127	0.3070	0.3843	0.3555	0.33
60977	FOODS_3_825_WI_3_evaluation	0.8340	0.6410	0.6562	0.6313	0.6987	0.7190	0.7840	0.71
60978	FOODS_3_826_WI_3_evaluation	1.0780	1.1870	1.0540	1.1150	1.2220	1.3870	1.2020	1.16
60979	FOODS_3_827_WI_3_evaluation	1.1240	1.0080	0.8184	1.0300	1.2810	1.1520	1.0700	0.94

30490 rows × 29 columns

In [ ]:
```
sub2 = sub.copy()
```

In [ ]:
```
sub2["id"] = sub2["id"].str.replace("evaluation$", "validation")
sub2
```

Out[71]:

	id	1942	1943	1944	1945	1946	1947	1948	19
30490	HOBBIES_1_001_CA_1_validation	0.8920	0.8190	0.8110	0.9043	1.1270	1.3020	1.3810	1.05
30491	HOBBIES_1_002_CA_1_validation	0.2659	0.2737	0.2620	0.2522	0.3071	0.3472	0.3040	0.21
30492	HOBBIES_1_003_CA_1_validation	0.4868	0.4275	0.4622	0.4670	0.5938	0.7036	0.6860	0.52
30493	HOBBIES_1_004_CA_1_validation	1.9570	1.5020	1.5750	1.7010	1.8630	2.3180	2.6700	2.13
30494	HOBBIES_1_005_CA_1_validation	1.4730	1.2610	1.1700	1.1080	1.3930	1.6680	1.5090	1.25
...	...	...	...	...	...	...	...	...	
60975	FOODS_3_823_WI_3_validation	0.5010	0.5205	0.5350	0.5020	0.5230	0.6855	0.6787	0.55
60976	FOODS_3_824_WI_3_validation	0.2323	0.2384	0.2540	0.3364	0.3499	0.3796	0.3757	0.32
60977	FOODS_3_825_WI_3_validation	0.8228	0.6220	0.6377	0.6045	0.6826	0.7610	0.8887	0.71
60978	FOODS_3_826_WI_3_validation	1.1470	1.2930	1.1380	1.1330	1.2000	1.3890	1.2960	1.18
60979	FOODS_3_827_WI_3_validation	1.3790	1.1210	0.9310	1.2430	1.4160	1.5300	1.4630	1.04

30490 rows × 29 columns

In [ ]:
```
sub3 = pd.concat([sub2, sub], axis=0, sort=False)
sub3
```

Out[72]:

	id	1942	1943	1944	1945	1946	1947	1948	19
30490	HOBBIES_1_001_CA_1_validation	0.8920	0.8190	0.8110	0.9043	1.1270	1.3020	1.3810	1.05
30491	HOBBIES_1_002_CA_1_validation	0.2659	0.2737	0.2620	0.2522	0.3071	0.3472	0.3040	0.21
30492	HOBBIES_1_003_CA_1_validation	0.4868	0.4275	0.4622	0.4670	0.5938	0.7036	0.6860	0.52
30493	HOBBIES_1_004_CA_1_validation	1.9570	1.5020	1.5750	1.7010	1.8630	2.3180	2.6700	2.13
30494	HOBBIES_1_005_CA_1_validation	1.4730	1.2610	1.1700	1.1080	1.3930	1.6680	1.5090	1.25
...	...	...	...	...	...	...	...	...	
60975	FOODS_3_823_WI_3_evaluation	0.5010	0.5205	0.5350	0.5020	0.5230	0.6855	0.6787	0.55
60976	FOODS_3_824_WI_3_evaluation	0.2323	0.2384	0.2540	0.3364	0.3499	0.3796	0.3757	0.32
60977	FOODS_3_825_WI_3_evaluation	0.8228	0.6220	0.6377	0.6045	0.6826	0.7610	0.8887	0.71
60978	FOODS_3_826_WI_3_evaluation	1.1470	1.2930	1.1380	1.1330	1.2000	1.3890	1.2960	1.18
60979	FOODS_3_827_WI_3_evaluation	1.3790	1.1210	0.9310	1.2430	1.4160	1.5300	1.4630	1.04

60980 rows × 29 columns

In [ ]:
```
sub3.columns = ['id'] + ['F' + str(c) for c in np.arange(1,29,1)]
```

In [ ]:
```
sub3.to_csv("Kaggke_LGBM_3r_March.csv",index=False)
```

Submission and Description	Private Score	Public Score	Use for Final Score
Kaggke_LGBM_3r_March.zip 42 minutes ago by srkef checking LGBM Model	0.56459	0.69091	☐

31	▲774	nagao	0.56364	44	8mo
32	▲3377	AjayNagar	0.56438	12	8mo
33	▲551	cjwh	0.56649	71	8mo

In [ ]:
```
# Got Rank Under 33.With Very Simple Approch in private LedgerBoard.
```

In [3]:
```
from prettytable import PrettyTable
t = PrettyTable(['Model_Number','Model_Name', 'Private Wrrmse Score','Public Wrr
t.add_row(['1', 'Base_Model',0.94398,1.00592])
t.add_row(['2', 'Random Forest Regressor',0.87519 ,0.78390])
t.add_row(['3','Extra Tree Regressor', .90082 ,0.81665])
t.add_row(['4', 'Lgbm',.56459 ,.69091])
print(t)
```

```
+--------------+-------------------------+----------------------+--------------
-------+
| Model_Number |        Model_Name       | Private Wrrmse Score | Public Wrrmse
Score |
+--------------+-------------------------+----------------------+--------------
-------+
|      1       |        Base_Model       |       0.94398        |       1.00592
|
|      2       | Random Forest Regressor |       0.87519        |        0.7839
|
|      3       |   Extra Tree Regressor  |       0.90082        |       0.81665
|
|      4       |           Lgbm          |       0.56459        |       0.69091
|
+--------------+-------------------------+----------------------+--------------
-------+
```